PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Philippe Armangau, et al.	
Serial No.: 10/603,951 Confirm 3116	Group Art Unit: 2113
Filed: 06/25/2003	Examiner: Wilson, Youlanda L
For: Data Recovery with Internet Protocol Replication With or Without Full Resync	Atty. Dkt. No.: 10830.096.NPUS0

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

Sir:

Please deduct the $500 fee of 37 C.F.R. 41.20(b)(2) for filing a brief in support of the appeal filed Nov. 21, 2006, from EMC Corporation Deposit Account No. 05-0889. A Fee transmittal form is submitted for this purpose.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, by virtue of an assignment recorded at Reel 014238 Frame 0365.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

Claims 1-36 have been presented for examination.

Claims 5, 6, 23, and 24 have been cancelled.

Claims 7-10 and 25-28 are allowed.

Claims 3, 15-18, 21, and 33-36 are objected to.

Claims 1, 2, 4, 11-14, 19, 20, 22, and 29-32 have been finally rejected, and are being appealed.

IV. STATUS OF AMENDMENTS

No amendment has been filed subsequent to final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The appellants' invention of claim 1 provides a method of recovery in an asynchronous remote copy system [FIG. 20] having a primary file system [451 in FIG. 20] at a primary site and a secondary file system [453 in FIG. 20] at a secondary site. The primary site becomes inoperative [step 462 in FIG. 21] during read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system [step 461 in FIG. 21]. The method includes responding [step 463 in FIG. 21] to the primary site becoming inoperative [step 462 in FIG. 21] by beginning read/write access to the secondary file system [step 474 in FIG. 23], making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system [step 473 in FIG. 23], and keeping a record of changes made to the secondary file system during the read/write access to the secondary file system [step 475 in FIG. 23]. Thereafter, when the primary site becomes operative [steps 464 and 465 in FIG. 21], the snapshot copy is used to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun [step 532 in FIG. 28], and then the changes made to the secondary file system during the read/write access to the secondary file system are written into the primary file system [steps 533 and 534 in FIG. 28]. The method further includes terminating read/write access to the secondary file system [step 466 in FIG. 21], and once the changes made to the secondary file system have been written into the primary file system [steps 541 and 542 in FIG. 30], restarting the read/write access to the primary file system [step 544 in

FIG. 30] and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system [step 543 in FIG. 30]. (Appellants' specification, page 3, lines 2-19.)

For example, appellants' FIG. 20, reproduced below, shows such an asynchronous remote copy system having a primary file system 451 at a primary site, a secondary file system 453 at a secondary site, and a replication service 450 for asynchronous remote copy of changes (e.g. delta sets) made to the priory file system 451 over an IP pipe 454 to the secondary file system 453. (See appellants' specification, page 42 line 7 to page 43 line 6.)

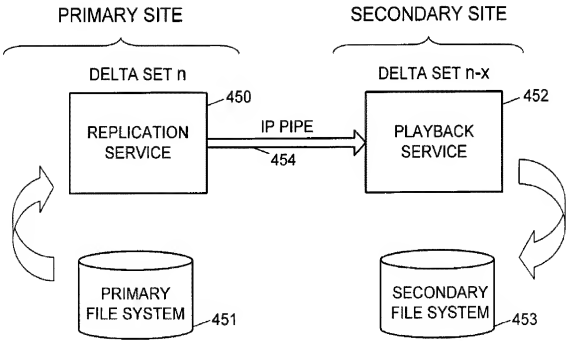


FIG. 20

FIG. 21, reproduced below, is a flowchart of the preferred recovery process. Initially, in step 461, data is replicated from the primary file system at the primary site by sending delta sets to the secondary file system at the secondary site. Then in step 462, the primary site becomes inoperative. In response, in step 463, the secondary site is activated for read/write access to the secondary file system. This is done by a subroutine for failover with a checkpoint and without sync, as further described below with respect to FIG. 23. Eventually, in step 464, the primary site becomes operative. In response, in step 465, the primary file system is synchronized to the state of the secondary file system, as further described below with respect to FIG. 28. Then in step 466, read/write access to the primary file system and replication of data from the primary file system to the secondary file system is resumed in a failback operation, as further described below with reference to FIG. 30. (Appellants' specification, page 43, lines 7-18.)

In step 473 of the failover operation of FIG. 23, reproduced below, the secondary site creates a snapshot copy of the "restart point" of the secondary file system. (Appellants' specification, page 44, line 4-8.) In step 474, the secondary file system is mounted as a read/write file system, and in step 475, the snapshot process retains and identifies all changes (delta) made to the secondary file system since the restarting point. (Appellants' specification, page 44, lines 13-15.)

In step 532 of the resync operation of FIG. 28, reproduced below, the primary site restores the primary file system to the state of the restarting point by obtaining a list of blocks from the save volume at the primary site, including the blocks in delta set n-x+1 to delta set n.

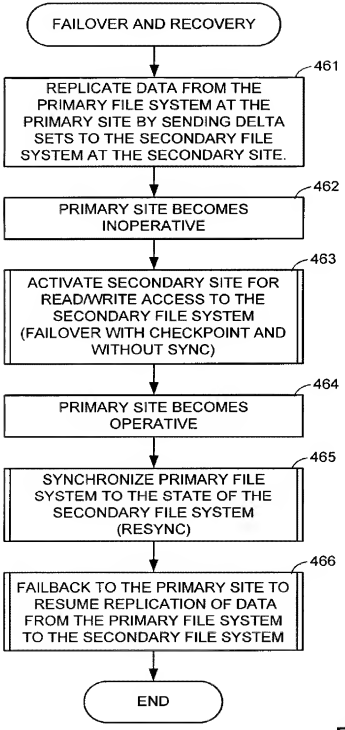

FAILOVER AND RECOVERY
461
REPLICATE DATA FROM THE PRIMARY FILE SYSTEM AT THE PRIMARY SITE BY SENDING DELTA SETS TO THE SECONDARY FILE SYSTEM AT THE SECONDARY SITE.
462
PRIMARY SITE BECOMES INOPERATIVE
463
ACTIVATE SECONDARY SITE FOR READ/WRITE ACCESS TO THE SECONDARY FILE SYSTEM (FAILOVER WITH CHECKPOINT AND WITHOUT SYNC)
464
PRIMARY SITE BECOMES OPERATIVE
465
SYNCHRONIZE PRIMARY FILE SYSTEM TO THE STATE OF THE SECONDARY FILE SYSTEM (RESYNC)
466
FAILBACK TO THE PRIMARY SITE TO RESUME REPLICATION OF DATA FROM THE PRIMARY FILE SYSTEM TO THE SECONDARY FILE SYSTEM
END

FIG. 21

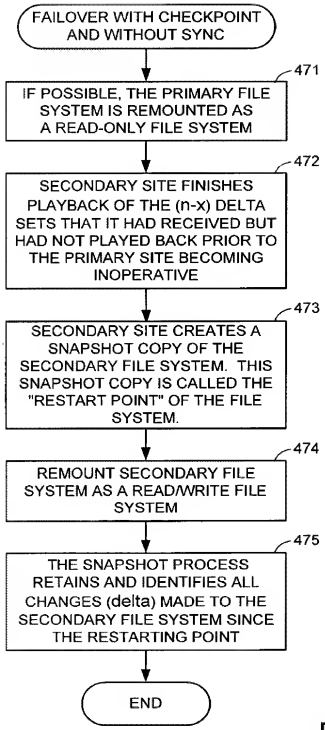

FAILOVER WITH CHECKPOINT AND WITHOUT SYNC
471
IF POSSIBLE, THE PRIMARY FILE SYSTEM IS REMOUNTED AS A READ-ONLY FILE SYSTEM
472
SECONDARY SITE FINISHES PLAYBACK OF THE (n-x) DELTA SETS THAT IT HAD RECEIVED BUT HAD NOT PLAYED BACK PRIOR TO THE PRIMARY SITE BECOMING INOPERATIVE
473
SECONDARY SITE CREATES A SNAPSHOT COPY OF THE SECONDARY FILE SYSTEM. THIS SNAPSHOT COPY IS CALLED THE "RESTART POINT" OF THE FILE SYSTEM.
474
REMOUNT SECONDARY FILE SYSTEM AS A READ/WRITE FILE SYSTEM
475
THE SNAPSHOT PROCESS RETAINS AND IDENTIFIES ALL CHANGES (delta) MADE TO THE SECONDARY FILE SYSTEM SINCE THE RESTARTING POINT
END

FIG. 23

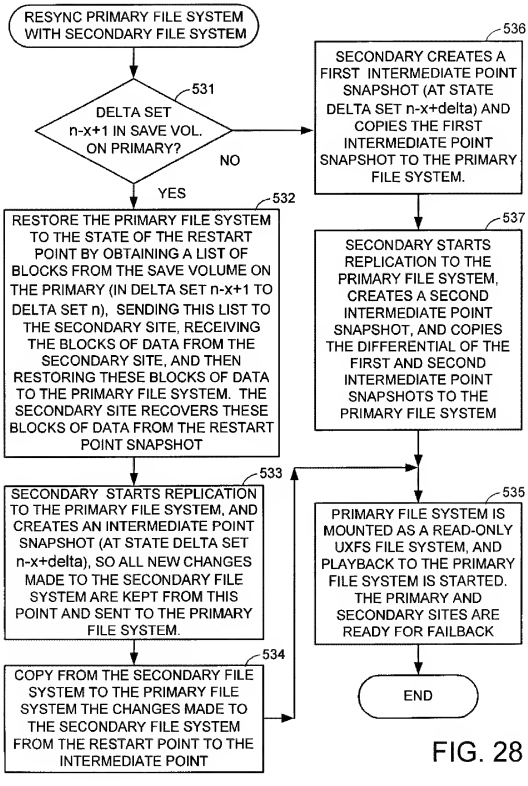

RESYNC PRIMARY FILE SYSTEM WITH SECONDARY FILE SYSTEM
531
DELTA SET n-x+1 IN SAVE VOL. ON PRIMARY?
NO
YES
536
SECONDARY CREATES A FIRST INTERMEDIATE POINT SNAPSHOT (AT STATE DELTA SET n-x+delta) AND COPIES THE FIRST INTERMEDIATE POINT SNAPSHOT TO THE PRIMARY FILE SYSTEM.
532
RESTORE THE PRIMARY FILE SYSTEM TO THE STATE OF THE RESTART POINT BY OBTAINING A LIST OF BLOCKS FROM THE SAVE VOLUME ON THE PRIMARY (IN DELTA SET n-x+1 TO DELTA SET n), SENDING THIS LIST TO THE SECONDARY SITE, RECEIVING THE BLOCKS OF DATA FROM THE SECONDARY SITE, AND THEN RESTORING THESE BLOCKS OF DATA TO THE PRIMARY FILE SYSTEM. THE SECONDARY SITE RECOVERS THESE BLOCKS OF DATA FROM THE RESTART POINT SNAPSHOT
537
SECONDARY STARTS REPLICATION TO THE PRIMARY FILE SYSTEM, CREATES A SECOND INTERMEDIATE POINT SNAPSHOT, AND COPIES THE DIFFERENTIAL OF THE FIRST AND SECOND INTERMEDIATE POINT SNAPSHOTS TO THE PRIMARY FILE SYSTEM
533
SECONDARY STARTS REPLICATION TO THE PRIMARY FILE SYSTEM, AND CREATES AN INTERMEDIATE POINT SNAPSHOT (AT STATE DELTA SET n-x+delta), SO ALL NEW CHANGES MADE TO THE SECONDARY FILE SYSTEM ARE KEPT FROM THIS POINT AND SENT TO THE PRIMARY FILE SYSTEM.
535
PRIMARY FILE SYSTEM IS MOUNTED AS A READ-ONLY UXFS FILE SYSTEM, AND PLAYBACK TO THE PRIMARY FILE SYSTEM IS STARTED. THE PRIMARY AND SECONDARY SITES ARE READY FOR FAILBACK
534
COPY FROM THE SECONDARY FILE SYSTEM TO THE PRIMARY FILE SYSTEM THE CHANGES MADE TO THE SECONDARY FILE SYSTEM FROM THE RESTART POINT TO THE INTERMEDIATE POINT
END

FIG. 28

The primary site sends this list to the snapshot copy facility at the secondary site. The secondary site retrieves the data of these blocks from the snapshot at the restart point, and returns the data to the primary site. The primary site receives these blocks of data and restores them to the primary file system. (Appellants' specification, page 49, lines 15-21.) In step 533, the snapshot copy facility starts replication to the primary file system, and creates an intermediate point snapshot (at state delta set n-x+δ), so all new changes made to the secondary file system since the intermediate point are kept and sent to the primary file system. However, they are not yet played back into the primary file system. (Appellants' specification, page 49 line 22 to page 50 line 2.) In step 534, the changes made to the secondary file system from the restart point to the intermediate point are copied from the secondary file system to the primary file system. (Appellants' specification, page 50, lines 5-7.) In step 535, the primary file system is mounted as a read-only UxFS file system, and playback to the primary file system is started. At this point, the resync is finished, and the primary and secondary sites are ready for failback. (Appellants' specification, page 50, lines 17-20).

In step 541 of the failback subroutine of FIG. 30, the primary file system is made almost identical to the secondary file system by the replication process. (Appellants' specification, page 53, lines 3-5.) In step 542, the secondary file system is re-mounted as read-only, and the last delta chunk is copied from the secondary file system to the primary file system in order to synchronize the primary file system from the secondary file system. (Appellants' specification, page 53, lines 6-9.) Then in step 543, there is a resumption of the replication of data from the primary file system and playback to the secondary file system. (Appellants' specification, page

53, lines 14-15.) Then in step 544, the primary file system is remounted as read/write. Therefore, the recovery process permits replication to be restarted as it was before the disaster. (Appellants' specification, page 53, lines 15-17.)

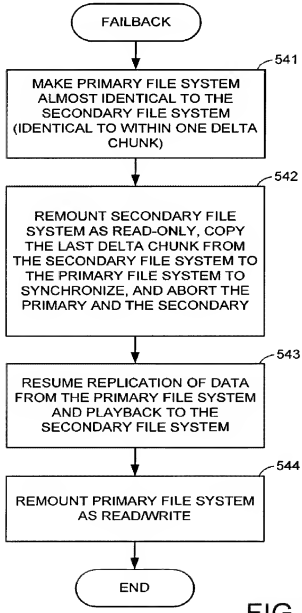


FIG. 30

Appellants' dependent claims 4 and 22 more particularly concerns the case where the state of the secondary file system existing when read/write access of the secondary file system was begun is a prior state of the primary file system existing before the primary site became inoperative. This case is tested for in the first step 531 of FIG. 28, and occurs when the delta set n-x+1 is in the save volume in the primary. In the first step 531 in FIG. 28, the delta set identifier (n-x) for the restart point is read from the delta set attribute of the restart point snapshot, and it is incremented by one to compute (n-x+1). The save volume at the primary site is searched for the delta set (n-x+1). Execution continues from step 531 to step 532 if the delta set n-x+1 is found in the save volume at the primary site. In this case, the primary site should also have all of the delta sets from delta set n-x+1 to delta set n, and the primary file system can be restored to the state of the restart point in step 532 by an "undo" of the data blocks of these delta sets. (Appellants' specification, page 48 line 22 to page 49, line 7.) In step 532, the primary site restores the primary file system to the state of the restarting point by obtaining a list of blocks from the save volume at the primary site, including the blocks in delta set n-x+1 to delta set n. The primary site sends this list to the snapshot copy facility at the secondary site. The secondary site retrieves the data of these blocks from the snapshot at the restart point, and returns the data to the primary site. The primary site receives these blocks of data and restores them to the primary file system. (Appellants' specification, page 49, lines 15-21.)

Appellants' independent claim 11 defines a method of recovery from a disruption at a primary site in an asynchronous remote copy system [FIG. 20] in which changes made to data blocks of a primary file system [451 in FIG. 20] at the primary site are transmitted to a secondary file system [453 in FIG. 20] at a secondary site. The primary site stores a list of the data bocks [in save volume 459 in FIG. 29] that have been changed in the primary file system. The method includes accessing the list of the data blocks that have been changed in the primary file system to restore the primary file system to a prior state at a restart point [step 532 in FIG. 28]. The prior state at the restart point includes changes made to the primary file system that have been transmitted to the secondary site. The primary file system is restored by determining from the list the data blocks that have been changed in the primary file system since the restart point, and obtaining from the secondary site the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point, and writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point. (Appellants' specification, page 5, line 16, to page 6 line 6; step 532 in FIG. 28; appellants' specification, page 49, lines 4-21.)

As shown in appellants' FIG. 29, reproduced below, a list of blocks in the delta sets n to n-x+1 is obtained from the save volume 459 at the primary site and sent to the snapshot copy facility 456 at the secondary site. The snapshot copy facility 456 returns "before images" of the requested blocks over an IP pipe 458 to the primary file system 451 to restore the primary file

system to the state of the restart point snapshot. (Appellants' specification, page 52, lines 9-16.)

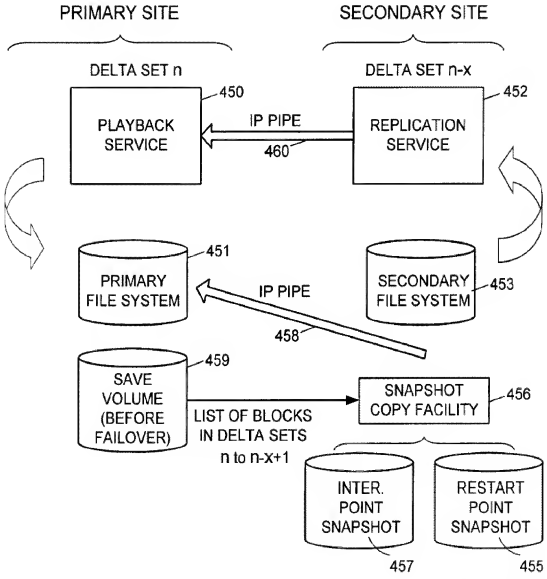


FIG. 29

As shown in FIG. 20, reproduced above, a disaster occurs when the playback service at the secondary site is playing back delta set (n-x) and replication service 452 at the primary site is replicating delta set (n). (Appellants' specification, page 42, lines 15-22.) The (n-x) deltasets are played back into the secondary file system 453, to put the secondary file system 453 into a state called a "restart point". (Appellants' specification, page 43, lines 20-21.)

As shown in step 532 of FIG. 28, reproduced above, the primary site restores the primary file system to the state of the restarting point by obtaining a list of blocks from the save volume at the primary site, including the blocks in delta set n-x+1 to delta set n. The primary site sends this list to the snapshot copy facility at the secondary site. The secondary site retrieves the data of these blocks from the snapshot at the restart point, and returns the data to the primary site. The primary site receives these blocks of data and restores them to the primary file system. (Appellants' specification, page 49 lines 15-21.) The primary file system is restored to the state of the restart point by an "undo" of the data blocks of these delta sets. The primary file system is restored with "before images" of these data blocks from the secondary site in order to "undo" the changes. (Appellants' specification, page 48, line 21 to page 49, line 14.)

Appellants' independent claim 19 defines an asynchronous remote copy system [FIG. 20] including a primary data storage system and a secondary data storage system. The primary data storage system has a primary file system [451], and the secondary data storage system has a secondary file system [453]. The primary data storage system is programmed for read/write access to the primary file system and asynchronous remote copy of changes made to the primary

file system being copied to the secondary file system. (Step 461 in FIG. 21.) The secondary data storage system is programmed to respond to the primary data storage system becoming inoperative during the asynchronous remote copy of changes made to the primary file system being copied to the secondary file system [step 462 in FIG. 21] by beginning read/write access to the secondary file system, making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system, and keeping a record of changes made to the secondary file system during the read/write access to the secondary file system [step 463 in FIG. 21; FIG. 23]. Moreover, the primary data storage system and the secondary data storage system are programmed for recovery when the primary data storage system becomes operative [step 464 in FIG. 21] by using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, and then writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system [step 465 in FIG. 21; FIG. 28], terminating read/write access to the secondary file system, and once the changes made to the secondary file system have been written into the primary file system, restarting read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system [step 466 in FIG. 21; FIG. 30]. (Appellant's specification, page 6, line 7, to page 7, line 5.) See the summary above for appellants' claim 1 for further references to the appellants' specification and figures.

Appellants' independent claim 29 defines an asynchronous remote copy system [FIG. 20] comprising a primary data storage system and a secondary data storage system, the primary data storage system having a primary file system [451 in FIG. 20] and the secondary data storage system having a secondary file system [453 in FIG. 20]. The primary data storage system is programmed for read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system. The primary data storage system stores a list of the data bocks [in save volume 459 in FIG. 29] that have been changed in the primary file system. The primary data storage system and the secondary data storage system are programmed for accessing the list of the data blocks that have been changed in the primary file system to restore the primary file system to a prior state at a restart point [step 532 in FIG. 28]. The prior state at the restart point includes changes made to the primary file system that have been transmitted to the secondary site. The primary file system is restored by determining from the list the data blocks that have been changed in the primary file system since the restart point, and obtaining from the secondary site the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point, and writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point. (Appellants' specification, page 5, line 16, to page 6 line 6; step 532 in FIG. 28; appellants' specification, page 49, lines 4-21.) See the summary above for appellants' claim 11 for further references to the appellants' specification and figures.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1, 4, 11-14, 19, 22, and 29-32 are unpatentable under 35 U.S.C. 102(e) as being anticipated by Leach et al. (U.S. Patent 6,694,447 B1).

2. Whether claims 2 and 20 are unpatentable under 35 U.S.C. 103(a) over Leach et al. (U.S. Patent 6,694,447 B1) in view of Martin et al. (U.S. Patent 6,016,501 A).

VII. ARGUMENT

1. Claims 1, 4, 11-14, 19, 22, and 29-32 are not unpatentable under 35 U.S.C. 102(e) and are not anticipated by Leach et al. (U.S. Patent 6,694,447 B1).

"For a prior art reference to anticipate in terms of 35 U.S.C. § 102, every element of the claimed invention must be identically shown in a single reference." Diversitech Corp. v. Century Steps, Inc., 7 U.S.P.Q.2d 1315, 1317 (Fed. Cir. 1988), quoted in In re Bond, 910 F.2d 831,15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (vacating and remanding Board holding of anticipation; the elements must be arranged in the reference as in the claim under review, although this is not an ipsis verbis test).

Leach et al. (U.S. Patent 6,694,447 B1) discloses a remote copy system including a primary server having a primary database at a primary site, and a secondary server having a secondary database at a secondary site. (Leach et al., FIG. 4A.) Leach et al. provide a method for increasing availability of an application during fail-back from the secondary site to the primary site following a failure at the primary site. The method includes copying data from active storage volumes to secondary storage volumes of the secondary site while the application runs on the secondary site and updates the active storage volumes. Once the secondary storage volumes of the secondary site are updated, the data is re-synchronized from the secondary

storage volumes of the secondary site to the primary storage volumes of the primary site. The steps of copying the data and resynchronizing the data are repeated for data updated by the application, during the resynchronization, until a time required to complete the resynchronization step for the updated data is within an acceptable downtime for the application. Once this step is complete, the application is failed-back to the primary site by bringing up the application at the primary site. (Leach et al., Abstract.)

In Leach et al., prior to the application failure [step 308 in Leach et al. FIG. 4C], the system replicates data generated by an application on the primary server to the secondary storage volumes of the secondary server. (Step 302 FIG. 4C.) The system generates a point-in-time image of data stored on the secondary storage volumes. (Step 304 in FIG. 4C.) The system stores the point-in-time image into a point-in-time storage volume at the secondary server. (Step 306 in FIG. 4C.)

Upon application failure (step 308 in FIG. 4C), execution continues to FIG. 5B. The system determines whether the secondary storage volume or the point-in-time storage volume contains the most current data. (Step 312 in FIG. 5B.) If the secondary storage volume contains the most current data, then the secondary storage volume point-in-time storage volumes are updated with the most current data using the steps in FIG. 5C. (Steps 313 and 314 in FIG. 5B; FIG. 5C.) Then the system fails over to the secondary server and the application runs from the secondary server. (Steps 328 and 330 in FIG. 5B.)

Once the primary server failure is corrected (step 332 in FIG. 5B), then data is copied from the active storage volumes to the secondary storage volumes while the application runs on

the secondary server (step 342 in FIG. 7B), and data is re-synchronized from the secondary storage volumes to the primary storage volumes on the primary server (step 344 in FIG. 7B). Once the time of step 342 is $\leq$ an acceptable application downtime (step 346 in FIG. 7B), then fail-back occurs to the primary server by bringing-up the application on the primary storage volumes. (Step 348 in FIG. 7B.) The application is terminated on the secondary server computer (step 352 in FIG. 7C), the system performs a final re-synchronization of data from the secondary storage volumes to the primary storage volumes (step 354 in FIG. 7C), and then the application is brought up on the primary storage volumes of the primary server (step 356 in FIG. 7C). Then the system returns to step 302 in FIG. 4C.

A comparison of the flowcharts in Leach et al. to the appellants' FIG. 21 in the table below shows that the system of Leach et al. performs high-level functions similar to those indicated in appellants' FIG. 21, but the system of Leach et al. performs the "failover," "resync," and "failback" functions in different ways.

HIGH-LEVEL FUNCTION	APPELLANTS' FIGURES	LEACH ET AL. FIGURES
Replication from primary to secondary	FIG. 21, step 461	FIG. 4C, step 302
Primary site becomes inoperative	FIG. 21, step 462	FIG. 4C, step 308
Failover from primary site to secondary site	FIG. 21, step 463 (FIG. 23)	FIG. 5B, steps 312-328
Primary site becomes operative	FIG. 21, step 464	FIG. 5B, step 332
Resync the primary site to the secondary site	FIG. 21, step 465 (FIG. 28)	FIG. 7B, steps 342-344
Failback to the primary site from the secondary site	FIG. 21, step 466 (FIG. 30)	FIG. 7B steps 348-356

Claims 1 and 19

With respect to appellants' independent claims 1 and 19, it is not seen where Leach et al. discloses: "making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system, and keeping a record of changes made to the secondary file system during the read/write access to the secondary file system; and thereafter, when the primary site becomes operative, using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, and then writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system ..." so that recovery can be achieved as further recited in the claim. Page 3 line 13 of the Final Official Action says: "The asynchronous remote copy of changes is the point-in-time image." However, Leach col. 7 lines 11-13 says: "Once the fail-over process is complete, the application runs on the secondary server 170 and updates the point-in-time storage volumes 190 as depicted in FIG. 6." Therefore the updating of the point-in-time storage volumes 190 by the application changes the state in the point-in-time snapshot volumes 190 so that the state at the beginning of the read/write access to the secondary file system is not preserved in the point-in-time storage volumes 190 for use in recovering the primary file system. In a similar fashion, the running of the application on the secondary server 170 causes blocks to be modified on the secondary storage volumes, as described in Leach col. 7 lines 18-21: "The secondary storage volumes (Instant Image Master Volumes) 188, are updated using the fast resynchronization process as described above. During normal operation, blocks that have been modified on the secondary storage volume are noted via

a scoreboard/bit map mechanism, contrary to the replication process of step 302." (Appellants' claim language itself calls for "changes made to the secondary file system during the read/write access to the secondary file system.")

Nor does Leach et al. disclose or need use of a snapshot copy of the secondary file system (at the beginning of read-write access to the secondary file system) for restoring the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun. Instead, Leach et al. restores the primary file system to a later state (including the changes made by the application running on the secondary server) using either a fast-resynchronization process or a full resynchronization from the secondary storage volumes 188 to the primary storage volumes 148 as described in col. 7 lines 50-57. "When the primary server's disks associated with the primary storage volumes 148 are still intact from before the disaster, only a fast resync, as described above, is required. Otherwise, a full resynchronization from the secondary storage volumes 188 to the primary storage volumes 148 is required." (Col. 7, lines 52-57.) As described in Leach et al. col. 6, lines 27-43, fast-resynchronization refers to a process that updates blocks as needed on a storage volume (e.g., blocks that have been modified on a source storage volume are immediately replicated to a target storage volume), in contrast to full-resynchronization in which all data of a storage volume are transferred. Thus, the scoreboard/bit map mechanism of Leach et al. col. 7 lines 18-21 would be used to indicate what storage blocks should be transferred from the secondary storage volumes 188 to the primary storage volumes 148 for the fast resync during failback when the primary storage volumes 148 are still intact from before the disaster.

Page 11 of the Final Official Action says:

> The 'using the snapshot copy to restore the primary file system...during the read/write access to the secondary file system...' is disclosed in [Leach et al.] column 7, lines 28-32, 50-57." As noted by Applicant, the fast re-synchronization is disclosed in column 6, lines 27-33. Although fast re-synchronization is described during normal operation, within the context of failing back to the primary server's database the changes made to the secondary server's database will be sent to the primary server's database since the primary server's database and the secondary serve's database were synchronized within the time of failure to the same point. Otherwise, all data on the secondary server's database will be sent to the primary server's database.

Appellants respectfully disagree. Leach et al. column 7, lines 28-38 say:

> At step 324, the application database is recovered. Finally, at step 326 archive logs from the secondary storage volumes 188 are applied to the active storage volumes 190. This process synchronizes the application database with the data contained in the primary storage volumes 148 prior to the failure. Although the method 318 for performing the updating operations of step 314 is described with reference to an application database, those skilled in the art will appreciate that additional data storage technologies, including but not limited to databases, file storage systems or other data storage means, are within the contemplation of the present invention.

This passage is describing ways of updating Leach et al.'s point-in-time storage volumes with the most current data. (See step 314 in FIG. 5B of Leach et al.) If the point-in-time storage

volumes, apparently 190 at the secondary in FIG. 5A, and the application database were synchronized with the data contained in the primary storage volumes 148 prior to the failure, then there would be no need to "restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun" as recited in appellants' claims 1 or 19 prior to using a "fast synchronization" or other means for "then writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system; ..." as further recited in appellants' claims 1 or 19.

Leach e t al. column 7, lines 50-57 say:

> At step 344, data is resynchronized from the secondary storage volumes 188 to the primary storage volumes 148 of the primary server 130. When the primary server's disks associated with the primary storage volumes 148 are still intact from before the disaster, only a fast resync, as described above, is required. Otherwise a full resynchronization from the secondary storage volumes 188 to the primary storage volumes 148 is required.

This passage is describing either a fast or full synchronization of the secondary storage volumes with the primary storage volumes on the primary server. (See step 344 in Leach et al. FIG. 7B.) In either case, this passage is not describing "using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun." "Using the snapshot copy [of the secondary file system] to restore the primary file system to the state of the secondary file system existing when read/write access of

the secondary file system was begun" is not necessary for the use of the "fast synchronization" because at the time of fail-over to the secondary server (step 328 in FIG. 5B) the point-in-time volumes have been synchronized with the data contained in the primary storage volumes 148 prior to the failure (step 314 in FIG. 5B) and the "fast synchronization" is used when the primary server's disks associated with the primary storage volumes 148 are still intact from before the disaster. In addition, keeping a record of changes made to the secondary file system during the read/write access to the secondary file system is different from making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system, and using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun is different from writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system.

In summary, appellants' claims 1 and 19 call for "in response to the primary site becoming inoperative ...beginning read/write access to the secondary file system, making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system, and keeping a record of changes made to the secondary file system during the read/write access to the secondary file system; ..." Appellants' claims 1 and 19 further call for "thereafter, when the primary site becomes operative, using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, and then writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system; ..."

There is no disclosure or suggestion that the resynchronization of data from secondary storage volumes to the primary storage volumes on the primary server in step 344 of Leach et al. FIG. 7B includes or requires using a snapshot copy [of the secondary file system existing when read/write access of the secondary file system was begun] to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, prior to writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system. A full resynchronization in step 342 of Leach et al. FIG. 7B does not use a snapshot copy of the secondary file system existing when read/write access of the secondary file system was begun, because the state of the active storage volumes in step 342 is different from the state of any snapshot copy of the secondary file system existing when read/write access of the secondary file system was begun, because the application running from the secondary server in step 330 of FIG. 5B has changed the state, and as required by appellants' claim language such changes are made to the secondary file system and are written into the primary file system after the restoration of the primary file system using the snapshot copy. (For example, in appellants' FIG. 28, a full resynchronization can be done in steps 536 and 537 without restoring the primary file system to the state of the restart point in step 532.) A fast resynchronization in step 342 of Leach et al. FIG. 7B does not use a snapshot copy of the secondary file system existing when read/write access of the secondary file system was begun, because the fast resynchronization is used when the primary storage volumes are still intact from before the disaster and therefore need not be restored with any snapshot copy of the secondary file system existing when read/write access of

the secondary file system was begun, prior to writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system.

Claims 4 and 22

With respect to appellants' claims 4 and 22, it is not seen where Leach et al. restores the primary file system to a state prior to the disruption of the primary server by the primary site keeping a list of blocks that have been changed in the primary file system during read/write access to the primary file system, and using the snapshot copy (of the secondary file system at the beginning of read/write access to the secondary file system) to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun by accessing the list of blocks that have been changed in the primary file system during the read/write access to the primary file system to determine the blocks that have been changed in the primary file system since said prior state of the primary file system.

Paragraph 7 on page 4 of the Final Official Action cites Leach et al. col. 6, lines 33-43 and col. 7, lines 39-67. It is not seen where these passages disclose the limitations of appellants' claims 4 and 22. Leach et al. col. 6, lines 33-43 simply describe replication, fast-resynchronization, and full-resynchronization. Leach col. 7, lines 39-67 describe fail-back in which "data is resynchronized from the secondary storage volumes 188 to the primary storage volumes 148 of the primary server 130. When the primary server's disks associated with the primary storage volumes 148 are still intact from before the disaster, only a fast resync, as described above, is required. Otherwise a full resynchronization from the secondary storage

volumes 188 to the primary storage volumes 148 is required." The fail-back of Leach et al. using a fast resync presumes that the state of the primary storage volumes 148 at the time of the disaster is the same as the state of the secondary file system existing when read/write access of the secondary file system was begun. Moreover, the appellants' method (of accessing the list of blocks that have been changed in the primary file system during the read/write access to the primary file system to determine the blocks that have been changed in the primary file system since said prior state of the primary file system) avoids a need for a full resync to restore the primary file system to a prior state.

Page 12, line 16 to page 13 line 2 of the Final Official Action say:

> The list of blocks is kept by the point-in-time database, which keeps track of changes made to the secondary server's database, as is what is described in column 6, lines 44-56. The fast re-synchronization, within the context of failing back to the primary server's database, the changes made to the secondary server's database will be sent to the primary server's database since the primary server's database and the secondary server's database were synchronized within the time of failure to the same point. Otherwise, all the data on the secondary server's database will be sent to the primary serve's database."

However, such a list of blocks kept by the point-in-time database for the fast resync is "a record of changes made to the secondary file system during the read/write access to the secondary file system" as recited in claims 1 and 19. Claims 4 and 22, which are dependent on claims 1 and 19, respectively, in addition recite "the primary site keeping a list of blocks that have changed in the primary file system during read-write access to the primary file system, ..."

Thus, as recited in claim 1 or 19, the restoration of the primary file system during the re-sync of the primary file system is a two-step process. The first step is using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun. The second step is writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system. The second step may use the record of changes made to the secondary file system during the read/write access to the file system. However, as recited in claim 4 or 22, the first step uses a different list of blocks that have changed in the primary file system during the read/write access to the primary file system. Specifically, in the first step, "the snapshot copy is used to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun by accessing the list of blocks that have been changed in the primary file system during the read/write access to the primary file system to determine the blocks that have changed in the primary file system since said prior state of the primary file system, and copying from the snapshot copy to the primary file system the blocks that have changed in the primary file system since said prior state of the file system."

Claims 4 and 22 are directed to the particular problem "wherein the state of the secondary file system existing when read/write access of the secondary file system was begun is a prior state of the primary file system existing before the primary site became inoperative, ..." This condition is avoided in Leach et al. since, as recognized by the Examiner, "the primary server's database and the secondary server's database were synchronized within the time of failure to the same point." This synchronization is done in step 314 of FIG. 5B of Leach et al. by updating the

application database or secondary so it is synchronized with data in the primary storage volumes 148 prior to the failure as shown in FIG. 5C of Leach et al. and described in Leach et al. col. 7 lines 23-38. Because Leach et al. update the PIT storage volumes with the most current data in step 314 of FIG. 5B for synchronization with the primary just prior to fail-over to the secondary server in step 328 of FIG. 5B and running the application from the secondary server in step 330 of FIG. 5B, there is no need for or suggestion of restoring the primary file system to a prior state before writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system.

Claims 11-14 and 29-32

With respect to appellants' independent claims 11 and 29, Leach is distinguished for similar reasons as given above for claims 4 and 22 for distinguishing the disclosure in Leach et al. col. 6, lines 33-43 and col. 7, lines 39-67. In addition, claims 11 and 29 recite that the prior state at a restart point (to which the primary file system is restored) includes changes made to the primary file system that have been transmitted to the secondary site. In other words, the primary file system is restored with changes that it once had, in order to restore the primary file system to the restart point. Moreover, claims 11 and 29 conclude with "writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point."

In effect, the primary file system is being restored to "undo" the changes in the primary file system since the restart point. However, this restoration is selective since it includes "the

primary site storing a list of the data blocks that have been changed in the primary file system" and "the primary file system being restored by determining from the list the data blocks that have been changed in the primary file system since the restart point, and obtaining from the secondary site the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point, ..."

Leach et al. fails to disclose the primary site storing a list of data blocks that have been changed in the primary file system so that it would be possible to determine from the list the data blocks that have been changed in the primary file system since the restart point. Nor does Leach et al. disclose obtaining from the secondary site the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point, and writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point. The fail-back of Lynch et al. using a fast resync presumes that the state of the primary storage volumes 148 at the time of the disaster is the same as the state of the secondary file system existing when read/write access of the secondary file system was begun, instead of a prior state at a restart point.

Leach et al. teaches an entirely different method of synchronizing a secondary server with a primary server at a restart point. Instead of restoring the primary to the state of the secondary at the restart point, Leach et al. in step 314 of FIG. 5B updates the secondary so that the secondary is synchronized with data in the primary storage volumes 148 prior to the failure as shown in FIG. 5C of Leach et al. and described in Leach et al. col. 7 lines 23-38. Because

Leach et al. update the PIT storage volumes with the most current data in step 314 of FIG. 5B for synchronization with the primary just prior to fail-over to the secondary server in step 328 of FIG. 5B and running the application from the secondary server in step 330 of FIG. 5B, there is no need for or suggestion of restoring the primary file system to a prior state by writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point.

Claims 12, 13, and 14 are dependent upon appellants' claim 11, and therefore are distinguished from Leach et al. by virtue of the limitations incorporated by reference from claim 11 that are not found in Leach et al., as discussed above.

Claims 30, 31, and 32 are dependent upon appellants' claim 29, and are and therefore are distinguished from Leach et al. by virtue of the limitations incorporated by reference from claim 29 that are not found in Leach et al., as discussed above.

2. Claims 2 and 20 are not unpatentable under 35 U.S.C. 103(a) over Leach et al. (U.S. Patent 6,694,447 B1) in view of Martin et al. (U.S. Patent 6,016,501 A).

The policy of the Patent and Trademark Office has been to follow in each and every case the standard of patentability enunciated by the Supreme Court in Graham v. John Deere Co., 148 U.S.P.Q. 459 (1966). M.P.E.P. § 2141. As stated by the Supreme Court:

> Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs,

> failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.

148 U.S.P.Q. at 467.

The problem that the inventor is trying to solve must be considered in determining whether or not the invention would have been obvious. The invention as a whole embraces the structure, properties and problems it solves. In re Wright, 848 F.2d 1216, 1219, 6 U.S.P.Q.2d 1959, 1961 (Fed. Cir. 1988).

Claims 2 and 20 are dependent upon claims 1 and 19, respectively, and therefore claims 2 and 20 incorporate by reference the limitations of claims 1 and 19, respectively, by virtue of 35 U.S.C. 112, fourth paragraph. Leach et al. fails to disclose certain limitations of claims 1 and 19, as set out above, and Martin et al. also fails to disclose these limitations. In particular, neither Leach et al. nor Martin et al. disclose or suggest using a snapshot copy [of the secondary file system existing when read/write access of the secondary file system was begun] to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, prior to writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system.

Where the prior art references fail to teach a claim limitation, there must be "concrete evidence" in the record to support an obviousness rejection. "Basic knowledge" or "common sense" is insufficient. *In re Zurko*, 258 F.3d 1379, 1385-86, 59 U.S.P.Q.2d 1693, 1697 (Fed. Cir. 2001).

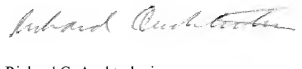
In addition, Leach et al. appears to be entirely satisfactory for its intended purpose of "increasing availability of an application during fail-back from a secondary site to a primary site following a failure at the primary site." (Leach et al., Abstract.) There is nothing other than the appellants' novel disclosure suggesting any improvement over the use of the "fast synchronization" of Leach et al. for re-sync of the primary when the primary volumes are intact after a disaster.

Hindsight reconstruction, using the applicant's specification itself as a guide, is improper because it fails to consider the subject matter of the invention "as a whole" and fails to consider the invention as of the date at which the invention was made. The critical inquiry is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination. In re Dembiczak, 175 F.3d 994, 999-1000, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999)(actual evidence and particular findings need to support the PTO's obviousness conclusion); In re Fritch, 972 F.2d 1260, 1266, 23 U.S.P.Q.2d 1780, 1784 (Fed. Cir. 1992)("It is impermissible to use the claimed invention as an instruction manual or 'template' to piece together the teachings of the prior art so that the claimed invention is rendered obvious."); Fromson v. Advance Offset Plate, Inc., 755 F.2d 1549, 1556, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985) (nothing of record plainly indicated that it would have been obvious to combine previously separate lithography steps into one process). See, for example, In re Gordon et al., 733 F.2d 900, 902, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984) (mere fact that prior art could be modified by turning apparatus upside down does not make modification obvious unless prior art suggests desirability of modification); Ex Parte Kaiser, 194 U.S.P.Q. 47, 48 (PTO Bd. of Appeals 1975) (Examiner's failure to indicate

anywhere in the record his reason for finding alteration of reference to be obvious militates against rejection).

In view of the above, the rejection of claims 1, 2, 4, 11-14, 19, 20, 22, and 29-32 should be reversed.

Respectfully submitted,



Richard C. Auchterlonie
Reg. No. 30,607

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, Suite 5320
Houston, TX 77002
713-751-0655

VIII. CLAIMS APPENDIX

The claims involved in this appeal are as follows:

1. A method of recovery in an asynchronous remote copy system having a primary file system at a primary site and a secondary file system at a secondary site, said method comprising:

in response to the primary site becoming inoperative during read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system, beginning read/write access to the secondary file system, making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system, and keeping a record of changes made to the secondary file system during the read/write access to the secondary file system; and thereafter,

when the primary site becomes operative, using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, and then writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system; and

terminating read/write access to the secondary file system, and once the changes made to the secondary file system have been written into the primary file system, restarting read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system.

2. The method as claimed in claim 1, wherein the asynchronous remote copy of changes made to the primary file system being copied to the secondary file system includes using the Internet Protocol to transmit the changes made to the primary file system over a data network between the primary site and the secondary site.

4. The method as claimed in claim 1, wherein the state of the secondary file system existing when read/write access of the secondary file system was begun is a prior state of the primary file system existing before the primary site became inoperative, and the method includes the primary site keeping a list of blocks that have been changed in the primary file system during read/write access to the primary file system, and the snapshot copy is used to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun by accessing the list of blocks that have been changed in the primary file system during the read/write access to the primary file system to determine the blocks that have been changed in the primary file system since said prior state of the primary file system, and copying from the snapshot copy to the primary file system the blocks that have been changed in the primary file system since said prior state of the primary file system.

11. In an asynchronous remote copy system in which changes made to data blocks of a primary file system at a primary site are transmitted to a secondary file system at a secondary

site, the primary site storing a list of the data bocks that have been changed in the primary file system, a method of recovery from a disruption at the primary site, said method comprising:

accessing the list of the data blocks that have been changed in the primary file system to restore the primary file system to a prior state at a restart point, the prior state at the restart point including changes made to the primary file system that have been transmitted to the secondary site, the primary file system being restored by determining from the list the data blocks that have been changed in the primary file system since the restart point, and obtaining from the secondary site the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point, and writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point.

12. The method as claimed in claim 11, which further includes the secondary site responding to the disruption by making a snapshot copy of the secondary file system at the restart point once all of the changes to the primary file system that have been transmitted to the secondary file system have been written into the secondary file system, and wherein the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point are obtained from the snapshot copy at secondary site.

13. The method as claimed in claim 11, which further includes the secondary site activating the secondary file system for read/write access once all of the changes to the primary

file system that have been transmitted to the secondary file system prior to the disruption have been written into the secondary file system.

14. The method as claimed in claim 11, which further includes the secondary site responding to the disruption by activating the secondary file system for read/write access, and wherein the state of the primary file system at the restart point is the state of the secondary file system when the secondary file system is activated for read/write access.

19. An asynchronous remote copy system comprising a primary data storage system and a secondary data storage system, the primary data storage system having a primary file system and the secondary data storage system having a secondary file system, the primary data storage system being programmed for read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system,

wherein the secondary data storage system is programmed to respond to the primary data storage system becoming inoperative during the asynchronous remote copy of changes made to the primary file system being copied to the secondary file system by beginning read/write access to the secondary file system, making a snapshot copy of the secondary file system at the beginning of read/write access to the secondary file system, and keeping a record of changes made to the secondary file system during the read/write access to the secondary file system; and

wherein the primary data storage system and the secondary data storage system are programmed for recovery when the primary data storage system becomes operative by using the snapshot copy to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun, and then writing into the primary file system the changes made to the secondary file system during the read/write access to the secondary file system, terminating read/write access to the secondary file system, and once the changes made to the secondary file system have been written into the primary file system, restarting read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system.

20. The asynchronous remote copy system as claimed in claim 19, wherein the primary data storage system is programmed to perform the asynchronous remote copy of changes made to the primary file system being copied to the secondary file system by using the Internet Protocol to transmit the changes made to the primary file system over a data network between the primary data storage system and the secondary data storage system.

22. The asynchronous remote copy system as claimed in claim 19, wherein the state of the secondary file system existing when read/write access of the secondary file system was begun is a prior state of the primary file system existing before the primary data storage system became inoperative, and wherein the primary data storage system is programmed to keep a list of blocks that have been changed in the primary file system during the read/write access to the

primary file system, and to restore the primary file system to the state of the secondary file system existing when read/write access of the secondary file system was begun by accessing the list of blocks that have been changed in the primary file system during the read/write access to the primary file system to determine the blocks that have been changed in the primary file system since said prior state of the primary file system, and copying from the snapshot copy to the primary file system the blocks that have been changed in the primary file system since said prior state of the primary file system.

29. An asynchronous remote copy system comprising a primary data storage system and a secondary data storage system, the primary data storage system having a primary file system and the secondary data storage system having a secondary file system, the primary data storage system being programmed for read/write access to the primary file system and asynchronous remote copy of changes made to the primary file system being copied to the secondary file system, the primary data storage system storing a list of the data bocks that have been changed in the primary file system;

wherein the primary data storage system and the secondary data storage system are programmed for recovering from a disruption in the asynchronous remote copy of changes made to the primary file system being copied to the secondary file system by accessing the list of the data blocks that have been changed in the primary file system to restore the primary file system to a prior state at a restart point, the prior state at the restart point including changes made to the primary file system that have been transmitted to the secondary data storage system, the primary

file system being restored by determining from the list the data blocks that have been changed in the primary file system since the restart point, and obtaining from the secondary data storage system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point, and writing into the primary file system the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point.

30. The asynchronous remote copy system as claimed in claim 29, wherein the secondary data storage system is programmed to respond to the disruption by making a snapshot copy of the secondary file system at the restart point once all of the changes to the primary file system that have been transmitted to the secondary file system have been written into the secondary file system, and wherein the secondary file system is programmed to obtain from the snapshot copy the data existing at the time of the restart point in the data blocks that have been changed in the primary file system since the restart point.

31. The asynchronous remote copy system as claimed in claim 29, wherein the secondary data storage system is programmed to activate the secondary file system for read/write access once all of the changes to the primary file system that have been transmitted to the secondary file system prior to the disruption have been written into the secondary file system.

32. The asynchronous remote copy system as claimed in claim 29, wherein the secondary data storage system is programmed to respond to the disruption by activating the secondary file system for read/write access, and the state of the primary file system at the restart point is the state of the secondary file system when the secondary file system is activated for read/write access.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.